# Recuperación parcial 2

Matemática Estructural y Lógica - Sección 5

Fecha límite: 26 de noviembre de 2017

## 1. Criptografía

La criptografía (del griego kryptós) consiste en el estudio y la práctica de técnicas que permiten la comunicación segura en presencia de adversarios terceros. En principio, con comunicación segura nos referimos a comunicación privada, en la cual se intercambian mensajes que personas no autorizadas no pueden leer. Esto se conoce como confidencialidad; y si bien es importante para la criptografía moderna, esta también se ocupa de otros aspectos de la seguridad de la información como la integridad, la autenticidad y el no repudio.

Para lograr la confidencialidad de un mensaje es importante entender el proceso de cifrado y descifrado. El proceso de cifrar consiste en convertir la información original (texto plano o plaintext, por ejemplo un mensaje $m$) en información incomprensible (texto cifrado o ciphertext). Descifrar es el proceso opuesto, y consiste en convertir esta información incomprensible de vuelta al mensaje original; o sea, a partir del texto cifrado recuperar el texto plano.

Un sistema de cifrado entonces consiste en una función de cifrado $f$ (que convierte el texto plano en texto cifrado) y una función de descifrado $f^{-1}$ (que toma el texto cifrado y lo descifra). Adicionalmente, la mayoría de sistemas de cifrado tienen un elemento adicional: una llave secreta $k$, que es un valor que comparten previamente las entidades que desean comunicarse de forma segura. En principio se trabaja con una llave, pero podría haber más de una, como veremos más adelante.

Las funciones $f$ y $f^{-1}$ usualmente no se aplican a todo el mensaje que se desea cifrar. El texto mas bien se divide en bloques, los cuales son cifrados individualmente para luego juntarse y ser enviados. Los bloques en principio son de letras del alfabeto (las que componen el mensaje), pero podrían ser también bloques de bits, que conforman una representación de bits del mensaje (en ASCII o UTF-8, por ejemplo).

Uno de los ejemplos más sencillos de un sistema de cifrado es el cifrado de César. El cifrado de un mensaje consiste en mover cada letra del mensaje $k$ posiciones hacia adelante en el alfabeto, devolviéndose al inicio en caso de llegar a última letra. De este modo, el cifrado utiliza bloques de tamaño 1 (una letra). La función de cifrado puede verse como:

$f(m) = (m + k) \mod 26$

donde $m$ es un número entre 0 y 25, que corresponde a la posición en el alfabeto de una letra del mensaje (0 corresponde a la A, 25 corresponde a la Z). La función para descifrar entonces sería

$f^{-1}(c) = (c - k) \mod 26$

donde $c$ es un número entre 0 y 25 y corresponde a la posición en el alfabeto de una letra del mensaje cifrado. Como es de esperarse, $k$ corresponde a la llave secreta, la cual es necesaria para cifrar y descifrar el mensaje.

## 2. Criptografía asimétrica y RSA

Una desventaja importante de cualquier sistema de cifrado simétrico (como los de la sección anterior, con una sola llave $k$) es que las personas que desean comunicarse deben previamente compartirse la llave secreta $k$, lo cual puede ser incómodo y además debe hacerse sobre un canal seguro. Este problema puede solucionarse mediante criptografía asimétrica: un esquema descubierto por primera vez por Clifford Cocks alrededor de 1973, mientras trabajaba para el GCHQ del gobierno británico. El algoritmo que Cocks descubrió era información clasificada, y fue en 1977 que una generalización de este fue descubierta independientemente por Ronald Rivest, Adi Shamir y Leonard Adleman. Por las iniciales de estos últimos el primer sistema de cifrado asimétrico se conoce como RSA.

A diferencia de los sistemas de cifrado simétricos, en un sistema de cifrado asimétrico cada participante cuenta con un par de llaves: una pública y una privada. La llave pública de un participante es conocida por todos aquellos que deseen comunicarse con él, mientras que la llave privada es conocida por él únicamente. En el caso de RSA la llave pública de un participante es una tupla de naturales $(n, e)$ y la llave privada corresponde a una tupla $(n, d)$. La pareja de llaves se genera de la siguiente forma:

i. Se escogen $p$ y $q$ dos primos y se calcula $n = pq$ su producto. $p$ y $q$ son extremadamente grandes (cada uno por lo general de 200 o más dígitos).

ii. Se escoge al azar un número $e$ que sea primo relativo con $(p - 1)(q - 1)$.

iii. Se calcula $d$ el inverso de $e$ módulo $(p-1)(q-1)$ (¿por qué debería existir?). Es decir, $d \cdot e \equiv_{(p-1)(q-1)} 1$.

El numeral ii. garantiza que siempre se puede calcular el inverso de $e$ (módulo $(p - 1)(q - 1)$). Este proceso se puede realizar fácilmente si se conocen $p$ y $q$, la factorización en primos de $n$. Esto quiere decir que la seguridad de RSA depende de la dificultad de factorizar en primos un número grande, lo cual, incluso con los mejores algoritmos hoy en día, tiene una complejidad exponencial con respecto a la longitud de $n$. Para un atacante que no conoce $p$ y $q$ es muy difícil averiguar $d$, la llave secreta de un participante.

Una vez se tienen las llaves, una persona que desee enviar un mensaje deberá conocer la llave pública del destinatario. Cuando el destinatario recibe el mensaje cifrado, él utiliza su llave privada para descifrarlo.

Para cifrar, llamaremos $M$ al mensaje que se desea enviar. A cada letra que compone el mensaje le asignamos un número de 2 dígitos, comenzando con 00 y terminando en 25. De esta forma convertimos nuestro mensaje en un gran bloque con una cantidad par de dígitos. Este gran bloque lo partimos en bloques de tamaño $N$ donde $N$ es el máximo número par tal que el número $(25 \dots 25)$ con $N$ dígitos es menor a $n$ (el $n$ de las llaves pública y privada). Hecho esto, aplicamos a cada bloque $m$ la siguiente función:

$c = m^e \bmod n$

y juntamos todos los bloques para obtener el mensaje cifrado $C$, el cual enviamos.

Para descifrar el texto cifado $C$, aplicamos a cada bloque $c$ la siguiente función

$m = c^d \bmod n$.

Por ejemplo, supongamos que queremos cifrar el mensaje "HOLA" y enviarlo a Alice. Alice tiene llave pública (2537,13) (si lo desea, verifique que esta llave cumple las propiedades mencionadas arriba, utilice que $2537 = 43 * 59$). Por otro lado, la llave privada de Alice es (2537,937) (pero esto no lo sabe nadie, solo Alice). Para cifrar el mensaje, convertimos las letras a dígitos:

07141100.

A continuación partimos este gran número en bloques, los bloques tendrán tamaño 4 porque $2525 < 2537$ (no podrían ser de tamaño 6 porque $2537 < 252525$). El resultado que obtenemos es

0714 1100.

Cada bloque lo elevamos a la 13 y le aplicamos módulo 2537. Para el primer bloque calculamos $714^{13}$, lo cual en principio puede parecer difícil, pero si recordamos que la suma y la multiplicación funcionan bien bajo operaciones de módulo, podemos utilizar la representación binaria del 13 (1101, o sea $13 = 8 + 4 + 1$) para realizar el cálculo:

$$\begin{aligned} 714^{13} \bmod 2537 &= 714^{8+4+1} \bmod 2537 = 714^8 \cdot 714^4 \cdot 714 \bmod 2537 \\ &= (714^2)^4 \cdot (714^2)^2 \cdot 714 \bmod 2537 \\ &= (2396^2)^2 \cdot 2396^2 \cdot 714 \bmod 2537 \\ &= 2122^2 \cdot 2122 \cdot 714 \bmod 2537 \\ &= 2246 \cdot 2122 \cdot 714 \bmod 2537 \\ &= 1191 \end{aligned}$$

De igual forma podemos cifrar el otro bloque y obtener $1100^{13} \bmod 2537 = 1906$. Entonces el mensaje cifrado $C$ que enviamos es:

1191 1906.

Cuando Alice recibe este mensaje, ella utiliza su llave privada para descifrar ($d = 937$). Para esto eleva cada bloque a la potencia 937 y halla el módulo 2537: $1191^{937} \bmod 2537 = 714$ y $1906^{937} \bmod 2537 = 1100$. Alice pega estos bloques y consigue 07141100 con lo cual recupera el mensaje original, "HOLA".

## 3. Programación del algoritmo

Usted debe desarrollar una aplicación que realize cifrado con RSA (en el lenguaje de programación de su preferencia) con las siguientes características:

1. La aplicación recibe $M$, $n$ y $e$. $M$ es un string no vacío (que tiene únicamente letras de la A a la Z, en mayúsculas, sin incluir la $\tilde{\text{N}}$). $n$ es un producto de dos primos $p$ y $q$ (de no más de 10 dígitos cada uno) y $e$ un entero positivo primo relativo con $(p-1)(q-1)$. Usted no debe generar estos 3 números, solo recibirlos como parámetro. Puede suponer que cumplen con todas las propiedades establecidas.

2. La aplicación convierte las letras del mensaje $M$ en bloques, aplica RSA a cada bloque y pega los bloques para retornar el mensaje cifrado $C$, como se describió más arriba.

3. El código de la aplicación debe incluir un método *exp* que recibe $a, b, n$ enteros como parámetro y devuelve $a^b \mod n$. Usted puede implementar este método como desee.

4. No se requiere que su aplicación tenga interfaz gráfica, podría correr únicamente en una línea de comandos.

Por ejemplo, si la aplicación recibe $M =$ "$HOLA''$, $n = 2537$, $e = 13$, debe retornar $C = 11911906$.

La aplicación será sustentada en horarios convenidos con el monitor, quien asignará una máxima nota de 0,6, la cual será sumada a la nota del segundo parcial sobre 5.

## 4. Bibliografía


1. Kenneth Rosen. Discrete Mathematics and Its Applications. 7a Edición. 2012 (capítulo 4.6).

2. Wikipedia. Cryptography. https://en.wikipedia.org/wiki/Cryptography